RECEIVED
CENTRAL FAX CENTER
MAR 13 2008

AMENDMENTS TO THE CLAIMS

1. (Currently amended) A method comprising:

saving a definition of a region in a program bounded by an entry breakpoint and an end breakpoint, wherein the entry breakpoint is executed conditionally;

saving a definition of a scoped breakpoint within the region;

if a thread that executes an instance of the program encounters the entry breakpoint, saving an identifier of the thread;

if the thread encounters the scoped breakpoint within the region, determining whether the identifier was saved in response to the thread that executes the instance of the program encountering the entry breakpoint;

if the identifier was saved in response to the thread that executes the instance of the program encountering the entry breakpoint and the scoped breakpoint was encountered by the thread, halting execution of the thread that encountered the scoped breakpoint; and

if the identifier was not saved, the thread that executes the instance of the program did not encounter the entry breakpoint, and the scoped breakpoint was encountered by the thread that executes the instance of the program, allowing execution of the thread to continue after the scoped breakpoint was encountered without giving control to a user. ~~determining whether a thread encountered a scoped breakpoint; and~~

~~if the determining is true, halting execution of the thread if the thread previously encountered an entry breakpoint.~~

2. (Currently amended) The method of claim 1, further comprising:

after the thread encounters the end breakpoint, removing the identifier of the thread that was saved. ~~if the determining is true, allowing execution of the thread to continue if the thread did not previously encounter the entry breakpoint.~~

3. (Currently amended) The method of claim 2, ~~claim 1,~~ further comprising:

PAGE 5/18 * RCVD AT 3/13/2008 9:48:54 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-4/10 * DNIS:2738300 * CSID:651 457 5622 * DURATION (mm-ss):06-34

allowing execution of the thread to continue upon the thread encountering the end breakpoint without giving control to the user. ~~wherein the scoped breakpoint is within a region bounded by the entry breakpoint and an end breakpoint.~~

4. (Currently amended) The method of ~~claim 1,~~ claim 3, further comprising:

allowing execution of the thread to continue upon the thread encountering the entry breakpoint without giving control to the user, ~~wherein the entry breakpoint is executed conditionally.~~

5. (Currently amended) An apparatus comprising:

means for saving a definition of a region in a program bounded by an entry breakpoint and an end breakpoint, wherein the entry breakpoint is executed conditionally;

means for saving a definition of a scoped breakpoint within the region;

means for saving an identifier of a thread that executes an instance of the program if the thread that executes the instance of the program encounters the entry breakpoint;

means for determining whether the identifier was saved in response to the thread that executes the instance of the program encountering the entry breakpoint if the thread encounters the scoped breakpoint within the region;

means for halting execution of the thread that encountered the scoped breakpoint if the identifier was saved in response to the thread that executes the instance of the program encountering the entry breakpoint and the scoped breakpoint was encountered by the thread that executes the instance of the program; and

means for allowing execution of the thread to continue after the scoped breakpoint was encountered without giving control to a user if the identifier was not saved, the thread that executes the instance of the program did not encounter the entry breakpoint, and the scoped breakpoint was encountered by the thread that executes the instance of the program. ~~means for determining whether a thread encountered a scoped breakpoint;~~

~~means for halting execution of the thread if the thread previously encountered an entry breakpoint and if the determining is true; and~~

PAGE 6/18 * RCVD AT 3/13/2008 9:48:54 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-4/10 * DNIS:2738300 * CSID:651 457 5622 * DURATION (mm-ss):06-34

~~means for allowing execution of the thread to continue if the thread did not previously encounter the entry breakpoint and if the determining is true.~~

6. (Currently amended) The apparatus of claim 5, further comprising:

means for removing the identifier of the thread that was saved after the thread encounters the end breakpoint.~~wherein the scoped breakpoint is within a region bounded by the entry breakpoint and an end breakpoint.~~

7. (Currently amended) The apparatus of ~~claim 5,~~ claim 6, further comprising:

means for allowing execution of the thread to continue upon the thread encountering the end breakpoint without giving control to the user.~~wherein the entry breakpoint is executed conditionally.~~

8. (Currently amended) The apparatus of ~~claim 6,~~ claim 7, further comprising:

means for allowing execution of the thread to continue upon the thread encountering the entry breakpoint without giving control to the user.~~and the end breakpoint.~~

9. (Currently amended) A ~~signal bearing~~storage medium encoded with instructions, wherein the instructions when executed comprise:

saving a definition of a region in a program bounded by an entry breakpoint and an end breakpoint, wherein the entry breakpoint is executed conditionally;

saving a definition of a scoped breakpoint within the region;

if a thread that executes an instance of the program encounters the entry breakpoint, saving an identifier of the thread;

if the thread encounters the scoped~~a scoped~~ breakpoint within the region, determining whether the identifier was saved in response to the thread that executes the instance of the program encountering the entry breakpoint;

if the identifier was saved in response to the thread that executes the instance of the program encountering the entry breakpoint and the scoped breakpoint was

PAGE 7/18 * RCVD AT 3/13/2008 9:48:54 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-4/10 * DNIS:2738300 * CSID:651 457 5622 * DURATION (mm-ss):06-34

encountered by the thread that executes the instance of the program, halting execution of the thread that encountered the scoped breakpoint; and

if the identifier was not saved, the thread that executes the instance of the program did not encounter the entry breakpoint, and the scoped breakpoint was encountered by the thread that executes the instance of the program, allowing execution of the thread to continue after the scoped breakpoint was encountered without giving control to a user.

10. (Currently amended) The ~~signal-bearing~~storage medium of claim 9, further comprising:

allowing execution of the thread to continue upon the thread encountering the entry ~~breakpoint; and~~ breakpoint without giving control to the user.

~~allowing execution of the thread to continue upon the thread encountering the end breakpoint.~~

11. (Currently amended) The ~~signal-bearing~~storage medium of claim 10~~claim 9~~, further comprising:

allowing execution of the thread to continue upon the thread encountering the end breakpoint without giving control to the user.

~~saving a definition of the scoped breakpoint within the region;~~

12. (Currently amended) The ~~signal-bearing~~storage medium of claim 11~~claim 9~~, further comprising:

after the thread encounters the end breakpoint, removing the ~~saved identifier~~identifier of the thread that was saved.

13. (Currently amended) A computer system comprising:

a processor; and

memory encoded with instructions, wherein the instructions when executed on the processor comprise:

PAGE 8/18 * RCVD AT 3/13/2008 9:48:54 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-4/10 * DNIS:2738300 * CSID:651 457 5622 * DURATION (mm-ss):06-34

saving a definition of a region in a program bounded by an entry breakpoint and an end breakpoint, wherein the entry breakpoint is executed conditionally,

saving a definition of a scoped breakpoint within the region,

if a thread that executes an instance of the program encounters the entry breakpoint, saving an identifier of the thread,

if the thread encounters the scoped breakpoint within the region, determining whether the identifier was saved in response to the thread that executes the instance of the program encountering the entry breakpoint,

if the identifier was saved in response to the thread that executes the instance of the program encountering the entry breakpoint and the scoped breakpoint was encountered by the thread, halting execution of the thread that encountered the scoped breakpoint, and

if the identifier was not saved, the thread that executes the instance of the program did not encounter the entry breakpoint, and the scoped breakpoint was encountered by the thread that executes the instance of the program, allowing execution of the thread to continue after the scoped breakpoint was encountered without giving control to a user.

14. (Currently amended) The computer system of claim 13, wherein the instructions further comprise:

after the thread encounters the end breakpoint, removing the identifier of the thread that was saved.

~~allowing execution of the thread to continue upon the thread encountering the entry breakpoint; and~~

~~allowing execution of the thread to continue upon the thread encountering the end breakpoint.~~

15. (Currently amended) The computer system of ~~claim 13,~~ claim 14, wherein the instructions further comprise:

PAGE 9/18 * RCVD AT 3/13/2008 9:48:54 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-4/10 * DNIS:2738300 * CSID:651 457 5622 * DURATION (mm-ss):06-34

allowing execution of the thread to continue upon the thread encountering the entry breakpoint without giving control to the user.~~after the thread encounters the end breakpoint, removing the saved identifier of the thread.~~

16. (Currently amended) The computer system of ~~claim 13,~~ claim 15, wherein the instructions further comprise:

allowing execution of the thread to continue upon the thread encountering the end breakpoint without giving control to the user.~~entry breakpoint is part of a conditional construct in the program.~~

17. (Currently amended) A method of configuring a computer, comprising:

configuring the computer to save a definition of a region in a program bounded by an entry breakpoint and an end breakpoint, wherein the entry breakpoint is executed conditionally;

configuring the computer to save a definition of a scoped breakpoint within the region;

configuring the computer to save an identifier of a thread that executes an instance of the program if the thread that executes the instance of the program encounters the entry breakpoint;

configuring the computer to determine whether the identifier was saved in response to the thread that executes the instance of the program encountering the entry breakpoint if the thread encounters the scoped breakpoint within the region;

configuring the computer to halt execution of the thread that encounters the scoped breakpoint if the identifier was saved in response to the thread that executes the instance of the program encountering the entry breakpoint and the scoped breakpoint was encountered by the thread that executes the instance of the program; and

configuring the computer to allow execution of the thread to continue after the scoped breakpoint was encountered without giving control to a user if the identifier was not saved, the thread that executes the instance of the program did not encounter the entry

PAGE 10/18 * RCVD AT 3/13/2008 9:48:54 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-4/10 * DNIS:2738300 * CSID:651 457 5622 * DURATION (mm-ss):06-34

breakpoint, and the scoped breakpoint was encountered by the thread that executes the instance of the program.

~~wherein the method comprises:~~

~~configuring the computer to determine whether a thread encountered a scoped breakpoint; and~~

~~configuring the computer to halt execution of the thread if the thread previously encountered an entry breakpoint and if the thread encountered the scoped breakpoint.~~

18. (Currently amended) The method of claim 17, further comprising:

configuring the computer to remove the saved identifier of the thread after the thread encounters the end breakpoint.~~configuring the computer to allow execution of the thread to continue if the thread did not previously encounter the entry breakpoint and if the thread encountered the scoped breakpoint.~~

19. (Currently amended) The method of claim 18, further comprising:

configuring the computer to allow execution of the thread to continue upon the thread encountering the end breakpoint without giving control to the user.~~claim 17, wherein the scoped breakpoint is within a region bounded by the entry breakpoint and an end breakpoint.~~

20. (Currently amended) The method of claim 19, further comprising:

configuring the computer to allow execution of the thread to continue upon the thread encountering the entry breakpoint without giving control to the user.~~claim 17, wherein the entry breakpoint is executed conditionally.~~

PAGE 11/18 * RCVD AT 3/13/2008 9:48:54 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-4/10 * DNIS:2738300 * CSID:651 457 5622 * DURATION (mm-ss):06-34